**Invention Disclosure Details**

**Name and Address of APPLICANT:**

|  |  |
| --- | --- |
| Name | Department/Address |
| **MANIPAL UNIVERSITY JAIPUR** | Jaipur-Ajmer Express Highway, Dehmi Kalan, Near GVK Toll Plaza, Jaipur, Rajasthan 303007 |

**Name and Address of Inventor:**

|  |  |
| --- | --- |
| Name | Department/Address |
| **Ayushman Gupta** | Student, Department of Artificial Intelligence and Machine Learning, School of Computer Science and Engineering, Manipal University Jaipur |
| **Dr. G. L. Saini** | Assistant Professor, Department of IoT and Intelligent Systems, School of Computing and Intelligence Systems, Manipal University Jaipur, India |

**DETAILS OF THE PUBLICATION/COMMUNICATION/EXHIBITION IN WHICH THE INVENTION WAS DISCLOSED PRIOR TO PATENT FILING IN PUBLIC DOMAIN.** (If YES kindly provide details)

**Response**: NO

**TITLE:**

**“Chronos” A Browsing History Categorization and Visualization Browser Extension**

**ABSTRACT**

1. There exist various software to track user browsing history and present it as a cluster of visited web addresses, while this can be effective for certain applications it is certainly not enough for the user to be able to analyse the kind of content they were consuming while they were on those websites. To solve this problem the proposed system suggests a Machine learning model packaged into a browser extension to classify the user browsing data into various categories and represent them in a visually appealing manner. This algorithm mainly consists of 2 development phases such as developing the Machine learning model which further consist of sub phases that data collection, data labelling, data preprocessing, implementation of standard classification algorithms and testing. The second development phase is the development of the browser extension and the integration of the two.

**THE PROBLEM TO BE SOLVED**

1. Chronos aims to equip users with a clear understanding of how they allocate their time online across various categories while browsing the internet on their personal computers.By providing the user with the information about their browsing habits the user can make informed decisions and retrospect on their time spent online throughout the day. It is not an easy task for a person to manually go through each entry in their browsing history to determine where they have been allocating their time throughout the day. Chronos aims to automate this process for them effortlessly and and in a visually appealing manner.
2. Chronos leverages browsers own history API to extract the user data with consent followed by the use of an advance deep learning algorithm paired with incremental learning to classify user browsing history data with at most precision and in a personalised manner. This data is then fed into the browser extension to show at in a visually appealing manner using charts and graphs.

**LITERATURE REVIEW**

1. The concept of analysing data with the help of machine learning algorithms to gain valuable insights isn’t a new one and user browsing history is no exception to that it is an area of active development and enhancements. Existing approaches however have used this data to identify the safety of URL’s rather than the genre classification.
2. **US20220046057A1 Inventors: Brody James Kutt, Peng Peng, Fang Liu, II William Redington Hewlett,** This patent focuses on deep learning for malicious URL classification. This is a prediction model whereas Chronos is a Categorisation model. The cited patent is a generic model and can not be personalised for users needs.
3. **US11609989B2 Inventors: Brian Sanford Jones, Zachary Mitchell Abzug, Jeremy Thomas Jordan, Giorgi Kvernadze, Dalian Quass,** This patent details a system in Uniform resource locator classifier and visual comparison platform for malicious site detection. This patent is used as a malicious URL detection model rather than genre categorisation model It offers organizational benefits but doesn't delve into the depth of content analysis and personalized insights that Chronos delivers.
4. **US10554736B2 Inventors : Seokkyung Chung, Farshad Rostamabadi, II William Redington Hewlett, Zhi Xu, Shadi Rostami-Hesarsorkh, Lin Xu, Lee Klarich,** This patent outlines a method for categorizing mobile URLs extracted from mobile applications. It utilizes a rule-based approach which depends heavily on the category of the mobile application itself. While useful for network security and policy enforcement, it is limited in scope and lacks the adaptability and user customization features of Chronos.

Chronos introduces the following significant developments in order to overcome the shortcomings of previous methods:

1. Classification Driven by Machine Learning

Chronos uses a machine learning model to classify websites based on their URLs and titles. Unlike the static rule-based technique in US10554736B2 this approach offers real-time categorization and allows the model to adapt and improve over time.

1. User-Centric Customization

Chronos gives users the ability to add, edit, and delete categories, allowing them to customize the categorization scheme to suit their own requirements and tastes. This degree of personalisation is not isn’t present in the cited patents.

1. User Refinement

Chronos allows users to manually reclassify URLs and learn from this new data to improve accuracy further. This feature is unique to Chronos and not found in existing solutions.

**THE OBJECTIVE OF THE INVENTION:**

1. Advantages for the users:

* Self-Awareness : Provides detailed insights into how users spend their time online, promoting mindful browsing habits.
* Personalized Insights: The chart visualization and trend analysis offer a clear understanding of browsing patterns, unique to each user's specific interests and activities.
* Data-Driven decision making : Allows users to make informed decisions about how they want to utilise their time online, potentially leading to increased productivity and reduced time wastage.
* Customizable Categories: Users can create their own categories, making the classification system personalized and relevant to their individual needs.
* User-Driven Accuracy: The manual reclassification feature empowers users to correct any misclassifications, ensuring the accuracy and relevance of the insights provided.
* Privacy-Focused: every bit of user data is stored on their personal computer ensuring privacy.

1. Technical Advantages :

* Efficient design : the product is lightweight and user efficient. Because the ML model is stored and run locally the cost of deployment is minimum.
* Adaptive learning : The model learns with more and more data and feedback from the user maximising the accuracy of over time.
* Each entry is classified in real time reducing the overall processing pressure on the system that would occur from
* Future scope for technical explansion – the algorithm has various other use cases ranging from employee productivity tracking, child activity tracking to a category based recommendation system.

1. Potential Societal Impact :

* Chronos contributes towards a bigger movement against unhealthy browsing and procrastination habits.
* By helping users understand their online behavior, Chronos could lead to increased productivity and more effective time management.

**SUMMARY OF THE INVENTION**

1. Core Functionality

Data Acquisition

* After the installation and user consent process is completed, Chronos retrieves the browsing history data through appropriate browser APIs.
* Regarding data collection and usage, the plugin complies with all privacy laws and user requests.

Machine Learning Integration

* Chronos uses a pre-trained machine learning model that is designed for classifying website titles and URLs.
* Using its internal representation of knowledge, the model guesses the category labels for every entry in browsing history.

Categorization

* The system classifies data into predefined categories mentioned further.

Data Aggregation

* Along with aggregating the browsing data chronos also tracks the amount of time spent on each URL.

Pie Chart Visualization

* The output is a self updating pie chart where each category is represented by a segment of the chart.
* The segment size is calculated by dividing the time spend on the category by the total browsing time in the specified time period.

Category Management

* A dedicated settings menu is implemented for personalisation
* The functionalities of the menu include adding, removing an renaming categories.

**Initial Dataset and Incremental Learning**

The precision in chrono’s skills lies in the user's own browsing history. Initially, it utilizes the user's personal dataset, categorized into the predefined classes of activities. This approach ensures accuracy and relevance, providing a solid foundation for subsequent ML attempts. Any miss classified entry is then corrected by the user itself in order to maximise accuracy.

As users engage with Chronos our data set enlarges, this data becomes an input for continual improvement. Through incremental learning techniques the model continuously updates and improves itself becoming more accurate with each use.

1. Advance Functionality

* Corrective Reclassification
  + The UI allows users to manually classify any data which the model is unable to process even creating new categories personalised to the user.
* Temporal Filtering
  + Users can sort the displayed data to analyse browsing patterns within specific time sets (e.g., last day, week, month).
  + This will allow user for a deeper understanding of their relationship with certain categories and time periods
* Trend Analysis
  + A line chart is implemented in order to identify and compare the trends between specified time periods and study the evolution of of user habbits.
  + Users will be able to study their usage of a selected category of interest to find out how much of its usage has increased or decreased.

1. Browser Integration

* Icon: The extension resides in the browser toolbar with a discreet but recognizable icon suggestive of time management or analysis.
* Pop-up Window: Clicking on the icon opens the extension's display window as a browser pop-up. This window should offer a responsive design for optimal viewing across device screen sizes.

1. Data Handling

* Local Storage: Chronos primarily uses the browser's local storage API to store the following:
  + Categorized browsing history entries
  + User-defined category settings
  + Customization for filtering or trend analysis preferences (if included)
* External Transmission (Limited): Chronos may offer an option to export summarized categorized history into a standard format (e.g., CSV, JSON) as an explicit user-initiated action.

1. Data Collection

The development of the ML categorization model utilizes a proprietary dataset. This dataset consists of the browsing history of the project's creators and their consenting associates. The data is collected and manually categorized in accordance with applicable privacy regulations into the following categories:

* **Search:** Search engines and query-related sites (Google, Bing, DuckDuckGo, Wikipedia)
* **­Social:** Social media platforms (Facebook, Instagram, Twitter, LinkedIn, Reddit)
* **News:** News sites and aggregators (CNN, BBC, New York Times)
* **Entertainment:** Streaming services, video sites, gaming sites (Netflix, Hulu, YouTube, Twitch)
* **Shopping:** E-commerce websites, online marketplaces (Amazon, eBay, Etsy)
* **Email:** Webmail providers (Gmail, Outlook, Yahoo Mail)
* **Finance:** Banking, investing, and financial information sites (Bank of America, Robinhood, Mint)
* **Travel:** Travel booking sites and informational resources (Expedia, Kayak, TripAdvisor)
* **Education:** Online courses, learning platforms, educational websites (Coursera, Khan Academy, university websites)
* **Hobbies:** Niche interests (birdwatching, rare plant collecting, historical reenactment)
* **Technology:** This is definitely a broad and important category. Websites about tech news, product reviews, software downloads, programming resources, etc. would fall into this. Examples: The Verge, CNET, GitHub, Stack Overflow.
* **Other:** This catches anything that doesn't fit neatly into the main categories, like niche interest websites etc.

1. **Preprocessing and Feature Engineering**

Text Cleaning

* Implement a thorough cleaning process to remove extraneous characters, normalize text (e.g., lowercase conversion), and address potential misspellings.
* Consider using regular expressions and established NLP libraries like NLTK.

Tokenization

* Experiment with word-level tokenization and potentially character-level n-gram tokenization for capturing partial word structures relevant to classifying certain websites.

Stop Word Removal

* Utilize a standard English stop word list, but consider allowing users to customize it for niche domains where common words might hold higher significance for classification.

Extracting Domain From URL

* “Urlparse” module is used from the standard python library

TF-IDF

* Implement TF-IDF vectorization using scikit-learn to compute the weighting scheme.
* Fine-tune the parameters controlling term frequency and inverse document frequency for optimal feature representation.
* 1-hot Encoding (Context-dependent):

Assess the suitability of 1-hot encoding. 1-hot encoding can be beneficial if the vocabulary size after tokenization is relatively limited and if preserving relative order between words is not essential for the classification task. Note that 1-hot encoding can drastically increase dimensionality, especially for large and diverse sets of browsing history data.

1. Candidate Classification Algorithms

Naive Bayes

* Implement a Multinomial Naive Bayes variant suited for text classification.
* Assess baseline performance due to its computational efficiency and relative robustness to overfitting.

Support Vector Machines (SVM)

* Experiment with linear and non-linear kernels (RBF) based on data characteristics.
* Potential for strong performance in text classification tasks due to its ability to find optimal decision boundaries.

Random Forest

* Evaluate a Random Forest ensemble classifier consisting of multiple decision trees.
* Random Forests often provide excellent generalization and handle high-dimensional data well.

Convolutional Neural Network (CNN)

* If computational resources permit, consider CNN architectures specifically tuned for text classification by utilizing 1D convolutions.
* CNNs excel at recognizing patterns and local structures in text, which might be beneficial for extracting discriminative features for browsing history classification.

1. Model Training & Evaluation

Dataset

* Employ a meticulously curated dataset with browsing history samples and manually assigned category labels.
* Dataset size should be sufficient for training potentially complex models (e.g., CNNs).
* Ensure a balanced representation of categories to prevent classification bias.

Data Division

* We stick with the conventional 80:20 splitting for the training and testing sets of data.

Performance Metrics:

* Accuracy, precision, recall, and F1-score, with a focus on per-category evaluation to identify weaknesses.
* Confusion matrix to determine results in detail.

Hyperparameter Optimization:

* Techniques like grid search and randomized search to fine-tune the parameters of each candidate model.
* regularization strategies to mitigate minimise overfitting.

1. Rationale for Algorithmic Complexity

For justifying the chosen ML model for Chronos We need to consider the trade-offs between:

* Model Accuracy: How well the model can correctly categorize browsing history entries.
* Computational Efficiency: How quickly the model can make predictions on new data.
* Explainability: How easily a human can understand the model's reasoning behind its classifications.

Several factors influence the complexity of a ML model

* Model Type: a Simpler approach like Naive Bayes is computationally efficient, while complex models like CNNs require more processing power.
* Dataset Size and Complexity: Large and diverse datasets benefit from more complex models because they can identify subtle patterns
* Real-Time Requirements: the program needs to update the pie chart in realtime with response to user interactions, a computational efficiency becomes essential as well along with accuracy.

1. Example Rationale for Algorithmic Complexity

Given the anticipated size of user browsing history data, a balance between accuracy and efficiency is crucial. While a complex model like a CNN might achieve the highest accuracy, the real-time performance requirements of the browser extension prioritize a more efficient model. We will initially explore models like Naive Bayes or Support Vector Machines (SVMs) with linear kernels. If performance falls short, we will consider a Random Forest while carefully assessing the impact on computational cost. Ultimately, the chosen model should strike a balance between delivering satisfactory accuracy for browsing history categorization and ensuring a responsive user experience within the browser extension.

1. Model Retraining and Adaptability

The extension will continuously learn from the user's browsing history data to personalize and improve the categorization model. Retraining will occur weekly, after a certain amount of new data has been acquired. The model will adapt incrementally. Secure local storage will be used for user-specific retraining data, with an option for contributing anonymized data to a centralized pool for enhanced model performance.

1. Guiding Principles

* Transparency: Chronos maintains transparency about data collection, usage, and storage, fostering user trust and ensuring compliance with regulations.
* User Control: Users retain ultimate control over their data, including explicit consent for accessing browsing history and choices for managing or deleting data.
* Data Minimization: Chronos collects and stores only essential data required for its intended functionality, minimizing the scope of potential risks.
* Security By Design: Security principles are integrated into the extension's architecture from inception, prioritizing secure coding practices and data handling.

1. Specific Mechanisms

* Upon installation, Chronos prominently displays a clear and concise permissions request informing users about the necessity of accessing browsing history data.
* Access to browsing history should only be granted after explicit user consent.
* Consider encrypting sensitive categorized browsing history data while stored locally, using robust algorithms (e.g., AES-256).
* Sensitive data should remain encrypted throughout its lifecycle whenever possible.
* Leverages the browser's designated mechanisms for secure local data storage, preventing unauthorized access.
* Any optional feature for exporting browsing data must implement a secure mechanism (e.g., HTTPS) to minimize interception risks.
* By default, no user data should be transmitted to external servers without an explicit, fully informed user action.

**CLAIMS**

1. A method for classification of browser history history entries using machine learning techniques included but not limited to:-

* retrieval of browsing history entries using browser history API
* tracking time spend on each url
* preprocessing retrieved data and extraction of titles and URLs
* implementing a machine learning model to classify data into predefined categories
* providing appropriate user interface to display categorised data in form of charts and graphs.

1. Visualising classified data comprising:-

* A machine learning model as described in above claim
* A user interface in the browser extension used to visualise data.
* A settings menu for the purpose of customisation by the users (adding, removing and editing categories)

1. A mechanism for the user to manually reclassify wrongly categorised entries.
2. Allowing users to filter the classified entries by time period (e.g., last day, week, month).
3. Identify and display trends of browsing behaviours
4. Integration of incremental learning behaviours.
5. A pie chart representing the time spent in each category;
6. A line chart showing trends in category usage over time.
7. The ability classify data into the following categories Search, Social, News, Entertainment, Shopping, Email, Finance, Travel, Education, Hobbies, Technology, Other

**FIGURE DESCRIPTIONS**

1. Fig1. The following figure is a level 0 data flow diagram. It is a very high level overview of the invention it depicts only the main interactions between the system an external entities.

Process: chronos

Dataflows: Browsing history data, User settings input, categorized data

1. Fig2. The following figure is a level 1 data flow diagram, it is medium level overview of the invention. The diagram includes :-

External entities – User, Browser

Processes – retrieve browsing data, preprocess data, Apply ML model, Categorise data, visualise data, manage settings, retrain ml model

Data stores : Stored data

1. Fig3. The following figure depicts a Level 2 data flow diagram, it is a low level depiction of the invention. It includes break down of details and and processes.

The diagram includes :-

External Entities: User, Browser, Processes, Preprocess Data, Apply ML Model, Postprocess Data, Categorize Data, Visualize Data, Manage Settings, Retrain ML Model, User Recategorization

Data Stores: Raw Data, Preprocessed Data, Classified Data, Processed Data, Stored Data, Updated Model, Updated Categories

Data Flows: Detailed data flows between sub-processes and data stores.

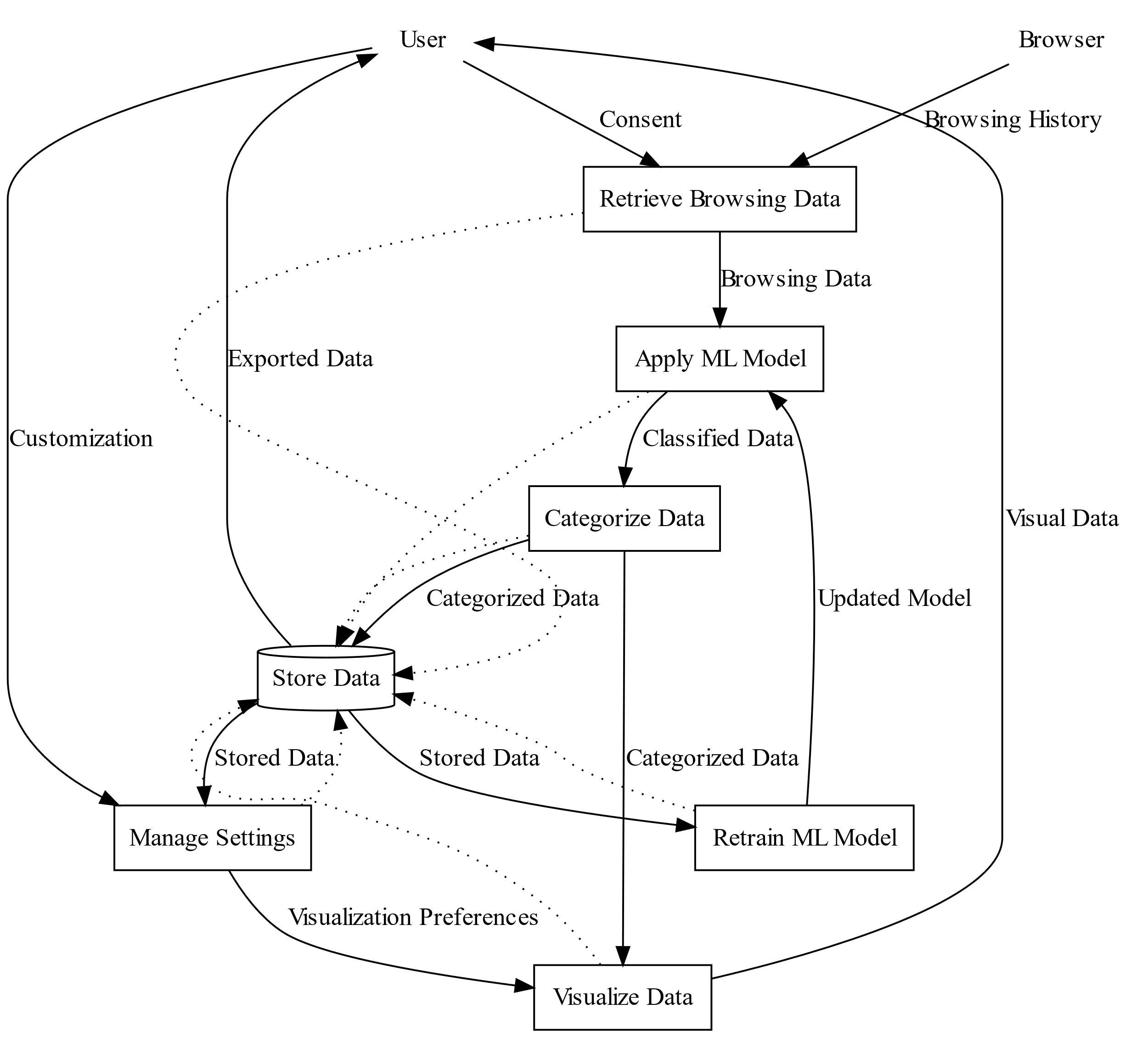
1. Fig 4. This figure depicts the development flow process of chronos and the basic understanding of the working on a higher level. The diagram includes: Data Collection, Data Preprocessing, Data Labeling, Data Encoding, Developing Machine Learning Model, Developing Browser Extension, Data Visualization.

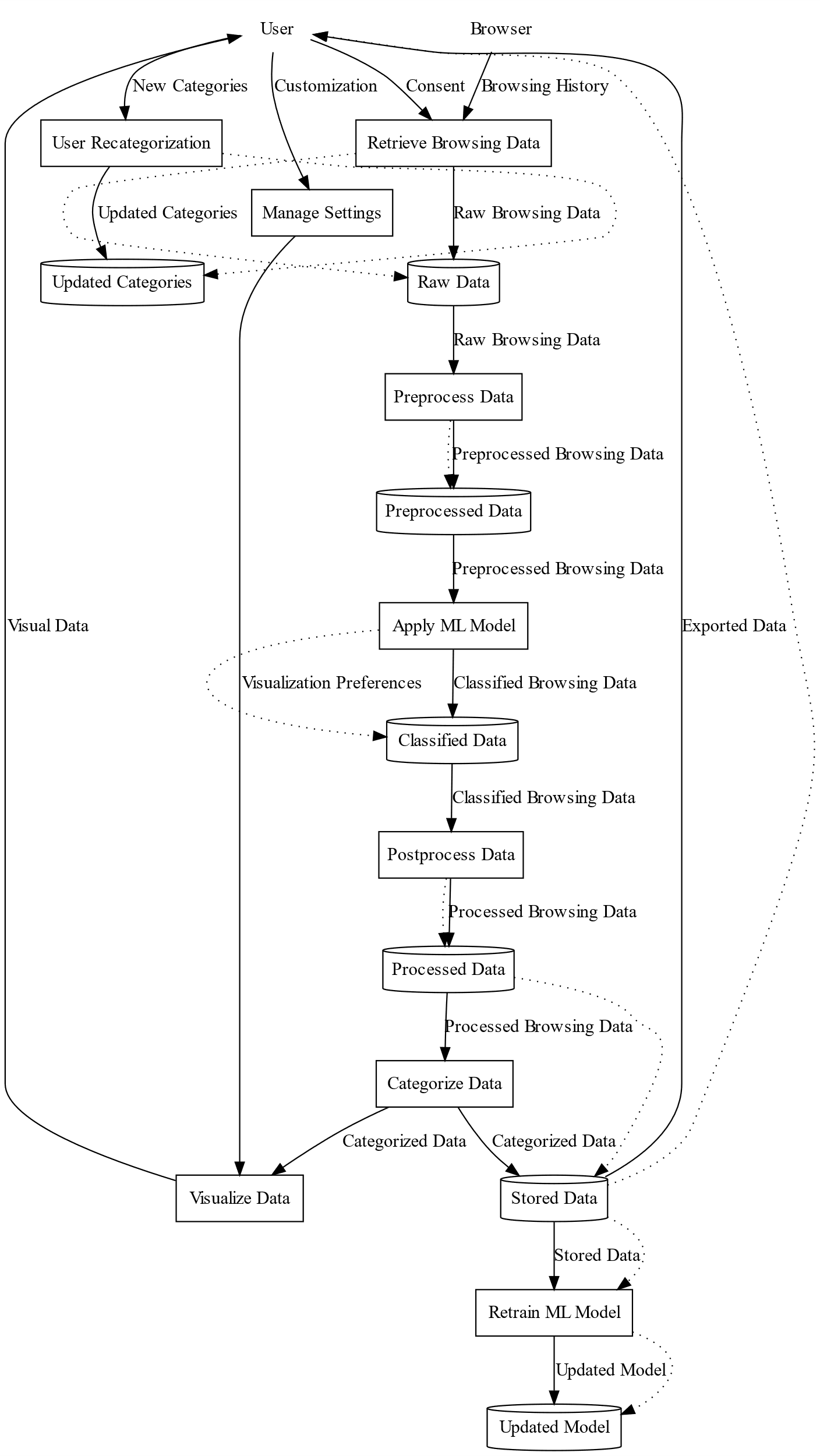
**FIGURES AND DIAGRAMS**

A diagram of a data flow

Description automatically generated

**Fig1. A level-0 DFD**

**Fig2. A level-1 DFD**



**Fig. 3 A level 2 DFD**

A diagram of a flowchart

Description automatically generated

**Fig.4 Flow of Development Process**